基于 SVM 的 Web 文本快速增量分类算法*

丁文军，薛安荣
（江苏大学 计算机科学与通信工程学院，江苏 镇江 212013）

摘 要：针对基于支持向量机的 Web 文本分类效率低的问题，提出了一种基于支持向量机 Web 文本的快速增量分类 FVI-SVM 算法。算法保留增量训练集中违反 KKT 条件的 Web 文本特征向量，克服了 Web 文本训练集规模巨大，造成支持向量机训练效率低的缺点。算法通过计算支持向量的共享最近邻相似度，去除冗余支持向量，克服了在增量学习过程中不断加入相似文本特征向量而导致增量学习的训练时间消耗加大、分类效率下降的问题。实验结果表明，该方法在保证分类精度的前提下，有效提高了支持向量机的训练效率和分类效率。

关键词：支持向量机；支持向量；最优分类超平面；KKT 条件；文本特征向量
中图分类号：TP393.09 **文献标志码**：A **文章编号**：1001-3695(2012)04-1275-04
doi:10.3969/j.issn.1001-3695.2012.04.020

Fast incremental learning SVM for Web text classification

DING Wen-jun, XUE An-rong
(*School of Computer Science & Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu* 212013, *China*)

Abstract: In Web text classification, with extremely large scale of the training set and the characteristic of changing rapidly, this paper proposed an algorithm named FVI-SVM based on incremental SVM for fast Web text classification. In order to conquer the problem of low efficiency of SVM which was aroused by a large scale of training set, datas in incremental training set which violate conditions of KKT would be exterminated. In order to conquer the problem of redundant support vectors which lead to the increasing of taining time consumption and decreasing of classification efficiency in incremental learning, exterminated the redundant support vectors by calculating shared nearest neighbors similarity. Experimental results show that the proposed method enhances the training and classification efficiency on a premise ensure the accuracy of classification.

Key words: support vector machine(SVM); support vector; optimal separating hyper-plane; Karush-Kuhn-Tucher(KKT); text feature vector

目前 Web 文本分类的增量学习算法主要有基于朴素贝叶斯(Naïve Bayes)、K 最近邻(K-nearest neighbor, KNN)、决策数(decision trees)、支持向量机(SVM)等。SVM 能有效处理高维稀疏和特征相关性大的问题，因此被广泛应用于文本分类。然而 Web 文本往往是按时间顺序的，变化快速且规模巨大，在初期获得一个完备的训练集是很困难的。因此，需要对 Web 文本进行有效的增量学习。

增量学习(incremental learning)是一种得到广泛研究和应用的智能化数据挖掘和知识发现技术。其思想是随着样本的不断积累，充分利用历史学习的结果，指导后续分类器的训练，学习精度也随之提高。基于结构风险最小化的 SVM 的分类性能仅依赖于支持向量，可以有效处理增量学习过程中对训练集产生过量匹配的问题。然而经典的 SVM 算法并不直接支持增量式学习，因此 SVM 增量式学习得到了广泛的关注。目前增量 SVM 算法主要有 Pronobis 等人[1]提出的利用整个增量训练集和历史训练集的并集作为当前训练集，将整个增量训练集加入当前训练集。由于训练 SVM 的时间复杂度为 $O(n^3)$，因此计算消耗非常大。针对这一问题，Duan 等人[2]和吴崇明等人[3]提出将违背 KKT 条件的新增样本集加入当前训练集。由于当前训练集中只有少部分样本违反 KKT 条件，因此该算法有效提高了训练效率。然而随着 SVM 增量学习的进行，训练集不断积累，在提高了分类精度的同时支持向量的数目也在不断增加。在 Web 文本增量学习过程中，支持向量集往往存在冗余现象。在分类过程中，测试样本需要与全部支持向量逐个进行点积运算，因此支持向量冗余将直接导致分类效率下降。另外，由于支持向量集合将被当成新的历史训练集，用于下一步增量学习，因此冗余支持向量的存在将增量训练 SVM 的计算消耗。为了去除冗余支持向量，提高增量学习效率，本文提出了一种 SVM 增量学习的文本分类算法(FVI-SVM)。

1 I-SVM 的 Web 文本分类

1.1 空间向量模型

目前文本的表示模型很多，其中最普遍使用的是向量空间模型(vector space model, VSM)[4]。在这种模型中，每篇文本被表示为一个特征向量，每个词语为该特征向量的一个维度。

$$x=\{x_1,x_2,\cdots,x_n\} \tag{1}$$

其中：n 是文档集合中不同词语的个数，x_i 表示词 t 在文本 d 中的权重。

TF-IDF[5]是 VSM 模型中一种常用的文本特征向量量化方

收稿日期：2011-09-10；**修回日期**：2011-10-18 **基金项目**：高校博士点基金资助项目(20093227110005)；校高级人才启动基金资助项目(09JDG041)；省科技型企业创新资金资助项目(BC2010172)

作者简介：丁文军(1986-)，男，湖南长沙人，硕士研究生，CCF 会员，主要研究方向为数据挖掘(787961271@qq.com)；薛安荣(1964-)，男，教授，博士，CCF 会员，主要研究方向为数据挖掘、信息安全.

法,它综合考虑单词在单个文档中出现的频率和该词语在文本集合中出现的频度。

$$x_i = \frac{tf(t,d) \times \log(N/n_t + 0.01)}{\sqrt{\sum [tf(t,d) \times \log(N/n_t + 0.01)]^2}} \tag{2}$$

其中:x_i 为词 t 在文本 d 中的权重;$tf(t,d)$ 为词 t 在文本 d 中的词频;N 为训练文本的总数;n_t 为训练文本集中出现 t 的文本数;分母为归一化因子。

1.2 I-SVM

VSM 的 Web 文本特征向量存在高维稀疏且特征相关性大的问题。SVM 可以有效处理高维问题且对特征相关性不敏感。因此,SVM 被广泛应用于基于 VSM 的文本分类。

SVM 基于结构风险最小化原理,通过最大化分类间隔,尽量提高学习机的泛化能力。其数学模型如下:

$$w(\alpha) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i \tag{3}$$

$$y_i(w\varphi(x_i) + b) \geqslant 1 - \xi_i \quad \xi_i \geqslant 0; i = 1,2,\cdots,l \tag{4}$$

其中:w 为超平面法向量;b 为超平面偏置;ξ_i 为松弛变量;C 为指定惩罚因子。

SVM 训练的求解最优分类问题,可以转换成下列二次函数的寻优问题:

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \tag{5}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad C > \alpha_i \geqslant 0; i = 1,2,\cdots,n \tag{6}$$

其中:$K(x_i \cdot x_j)$ 表示核函数,α_i 表示拉格朗日乘子。解该问题后得到最优分类函数:

$$y = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i K(x, x_i) + b\right) \tag{7}$$

虽然 SVM 在基于 VSM 模型的文本分类领域得到了广泛应用和较好的分类效果,但是 Web 文本是快速变化的且其规模是海量的,因此需要对 Web 文本进行有效的 SVM 增量学习(I-SVM)。其步骤如下:

a)将增量训练集加入历史训练集形成当前训练集;

b)在当前训练集上训练 SVM 分类器;

c)训练 SVM 产生的支持向量集作为新的历史训练集;

d)如果有新的增量训练集加入,转步骤 a)。

2 VI-SVM 的 Web 文本分类

目前 I-SVM 在 Web 文本分类中主要存在下列问题:

a)SVM 的训练时间复杂度为 $O(n^3)$。Web 文本具有海量特性,增量训练集的规模可能非常大,因此需要一种有效的增量训练集淘汰算法,降低训练 I-SVM 的计算消耗。

b)由于支持向量集合将被当成新的历史训练集,用于下一步增量学习,因此冗余支持向量的存在将增加训练分类器的计算消耗。另外,测试 Web 文本特征向量需要与全部支持向量进行点积运算,因此 SVM 的支持向量数目越多,SVM 的分类速度越慢。Web 文本特征向量规模很大,增量学习过程中,支持向量冗余不可避免,缺少有效的冗余支持向量修剪算法,直接限制了 I-SVM 在 Web 文本分类中的应用。

为了提高 I-SVM 的训练效率,本文提出了一种 Web 文本增量训练集淘汰算法 VI-SVM。为了排除由于支持向量冗余而造成 VI-SVM 训练和分类效率下降,提出了 FVI-SVM 算法。

增量训练集中潜在支持向量的引入将导致最优分类超平面发生改变,而潜在非支持向量的引入不能改变最优分类超平面。如图 1 所示,由于新增文本特征向量的加入,最优分类超平面由 F_1 改变为 F_2。其中,支持向量 B 的引入,改变了最优分类超平面,而非支持向量 A 的引入,没有改变最优分类超平面。因此,需要找到增量训练集中不能引起最优分类超平面发生改变的文本特征向量,并且从增量训练集中删除这一部分特征向量,以提高 SVM 训练效率。

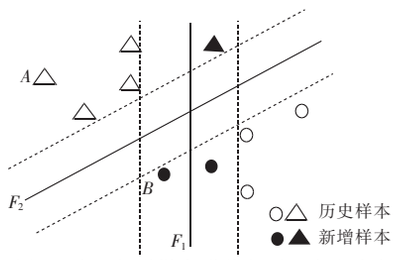

图 1 由于新增样本,分类超平面发生改变

为了找到增量训练集中引起最优分类超平面发生改变的文本特征向量,需要考察如下 KKT 条件:

在式(5)中,当且仅当对于每一个 x_i 都满足下列 KKT 条件时,如式(8)所示,α_i 才具有式(5)中二次规划的最优解。

$$\begin{cases} a_i = 0 \Leftrightarrow y_i f(x_i) > 1 \\ 0 < a_i < C \Leftrightarrow y_i f(x_i) = 1 \\ a_i = C \Leftrightarrow y_i f(x_i) < 1 \end{cases} \tag{8}$$

其中:α_i 表示拉格朗日乘子;C 为指定惩罚因子;y_i 为文本特征向量的类别,其取值为 $\{+1, -1\}$。$\alpha_i = 0$ 对应的文本特征向量为非支持向量,分布在最优分类超平面外,对最优分类超平面的构造没有影响;$0 < \alpha_i < C$ 表示对应文本特征向量位于最优分类超平面上,为支持向量;$\alpha_i = C$ 表示对应文本特征向量位于最优分类超平面间隔内,甚至被错分。

若新增文本特征向量刚好是原 SVM 的支持向量,即 $y_i f(x_i) = 1$,该新增文本特征向量刚好落在原 SVM 最优分类超平面上,不影响增量学习后的最优分类超平面;若新增文本特征向量远离最优分类超平面,同样不影响分类最优超平面;若新增文本特征向量落在最优分类超平面内部甚至被错分,即 $y_i f(x_i) < 1$,则表示最优分类超平面无法最优化分类问题,如果该新增文本特征向量不是异常,此时需要调整最优分类超平面。

另外,如果某个新增文本特征向量满足 $y_i f(x_i) = 1$,尽管最优分类超平面不会发生改变,但该特征向量携带分类信息,应该被保留,加入当前训练集。

由上述分析可以知道,当新增文本特征向量满足式(9)时,会导致最优分类超平面发生改变,因此这些文本特征向量将被保留;若不满足式(9)时,这些文本特征向量不能引起最优分类超平面发生改变,因此需要将这些文本特征向量从增量训练集中删除,以降低训练 SVM 的计算消耗。

$$y_i f(x_i) \leqslant 1 \tag{9}$$

3 FVI-SVM

尽管 VI-SVM 通过式(9)可以去除增量训练集中大量的非支持向量,提高 SVM 的训练效率,然而,随着增量学习的不断进行,大量相似的 Web 文本被用于训练 SVM 分类器,将形成大量的冗余支持向量。由于支持向量集合将被当成新的历史训练集,用于下一步增量学习,因此冗余支持向量的存在将增

加训练 SVM 的计算消耗。另外,在 SVM 分类过程中,如式(7)所示,测试文本特征向量需要与全部支持向量逐个进行点积运算,因此支持向量数目成为影响分类速度的主要因素,支持向量数目越多,计算量越大。

FVI-SVM 在 VI-SVM 的基础上,提出了一种剔除冗余支持向量的方法。为了剔除冗余支持向量,可以将式(7)改写成式(10),其中 M 表示非冗余的支持向量集合,L 表示冗余的支持向量集合。找到并删除 L 可以有效提高 SVM 分类效率。

$$y = \mathrm{sgn}\left(\sum_{i=1}^{M}\alpha_i K(x,x_i) + \sum_{i=1}^{L}\alpha_i K(x,x_i) + b\right) \tag{10}$$

其中:$M+L=N$,N 表示全部支持向量。

由于 L 的形成是因为 Web 文本中存在相似文本,因此,可以计算支持向量之间的相似性:相似性越高,表示其对应的 Web 文本越相似,越有可能是冗余的支持向量。可以通过计算 SNN 相似性度量支持向量之间的相似性。

定义 1　SNN 相似度。假定 p 和 q 是两个文本特征向量,则它们的相似度可以定义如下:

$$\mathrm{similarity}(p,q) = \mathrm{size}(NN(p) \cap NN(q)) \tag{11}$$

其中:$NN(p)$ 和 $NN(q)$ 分别对应 p 和 q 的最近邻居。

设 N 中支持向量(SV_i, SV_j)的 SNN 相似度值为 SNNValue,将(SV_i, SV_j)按照 SNNValue 由大到小排序并加入集合 List。由于增量学习而增加的支持向量数目为 SVCount。

从 List 中选择前 SVCount 个元素,将其中 SV_i 和 SV_j 分别加入 M 和 L。L 中支持向量将被当成冗余支持向量剪除;M 被保留,用于 SVM 分类。

加入 L 的冗余支持向量数目和由于增量学习而增加的支持向量数目均为 SVCount。由于增量学习而新增的支持向量数目等于被删除的冗余支持向量数目,因此支持向量集 N 中支持向量数目保持在一个稳定的规模。由式(7)可知,当支持向量数目不变时,算法的计算消耗保持不变,因此该算法可以保持稳定的分类效率。

4　FVI-SVM 的 Web 文本分类步骤

4.1　FVI-SVM 的 Web 文本分类

设增量训练集为 $X=\{x_1, x_2, \cdots, x_n\}$,其中 x_i 表示 Web 文本特征向量。历史训练集为 N_s^i,由上步增量学习中训练 SVM 产生的支持向量组成。当前训练集由 N_c^{i+1} 表示,其初始值为空。在 N_c^{i+1} 上训练 SVM 后其中支持向量组成的集合为 N_s^{i+1}。具体算法步骤如下:

输入:X, N_s^i, N_c^{i+1}。

输出:N_s^{i+1}。

a)将历史训练集全部加入当前训练集 $N_c^{i+1}=N_s^i$;

b)记录历史训练集 N_s^i 中支持向量数目,其大小为 SVCount1;

c)从新增训练集 X 中选择一个文本特征向量 $p(x_i, y_j)$,检查该特征向量是否违反 KKT 条件,即满足式(9),如果不违反,转步骤 d),否则转步骤 e);

d)将 p 从增量训练集中删除 $X=X-p$,转步骤 c);

e)直接将该违反 KKT 条件的文本特征向量加入当前训练集 $N_c^{i+1}=N_c^{i+1}+p$,并将 p 从增量训练集中删除 $X=X-p$;

f)如果 $X==\varphi$,则转步骤 g),否则转步骤 c);

g)在当前训练集 N_c^{i+1} 上训练 SVM 分类器,得出由支持向量构成的集合 N_s^{i+1};

h)记录 N_s^{i+1} 包含的支持向量数目,其大小为 SVCount2;

i)计算 N_s^{i+1} 中所有文本特征向量的 SNN 相似度,并按 SNN 相似度由大到小排列,加入集合 List。

j)将 List 包含的前 SVCount2 - SVCount1 对支持向量分别加入集合 L 和 M;

k)令 $N_s^{i+1}=N_s^{i+1}-L$;

l)利用 N_s^{i+1} 中支持向量对测试集进行分类。

4.2　算法分析

增量训练集 X 中包含 n 个文本特征向量,如果直接在 X 上训练 SVM 分类器,其时间复杂度为 $O(n^3)$。采用本文算法,根据式(9)计算是否违反 KKT 条件的时间复杂度为 $O(n\times l\times k)$,其中 l 表示支持向量数目,k 表示文本特征向量维度。在违反 KKT 条件的样本上训练 SVM 的时间复杂度为$O(n'^3)$,其中,n'表示违反 KKT 条件新增文本特征向量数目。在支持向量集 N 中求 SNN 相似度,需要查找所有支持向量的 K 近邻,其时间复杂度为 $O(l^2)$。因此,本文算法的训练时间复杂度为 $O(n\times l\times k)+O(n'^3)+O(l^2)$。由于 n'、l、k 都远远小于 n,因此本文算法可以有效提高训练效率。

原支持向量集由两部分组成:$N=M+L$,其中 M 表示非冗余支持向量,L 表示冗余支持向量。M 中支持向量数目为 l_1,L 中支持向量数目为 l_2。测试集中文本特征向量数目为 t,则原来分类的时间复杂度为 $O(t\times(l_1+l_2)\times k)$。本文提出一种修剪支持向量算法,去除冗余支持向量 L,其时间复杂度为 $O(t\times l_1\times k)$。由于 SVM 增量学习是一种持续学习算法,不断有新增文本特征向量加入用于训练 SVM 分类器,因此 l_2 的数值将不断增大,最后将远远大于 l_1。因此本文算法可以有效提高增量 SVM 的分类效率。

由于冗余支持向量产生的原因是存在大量的相似 Web 文本,去除冗余支持向量不会对 SVM 的分类精度产生影响。本文采用 SNN 相似度度量文本特征向量之间的相似性,能有效地找出冗余支持向量。

5　实验分析

5.1　实验数据集

本实验采用由李荣陆老师提供的复旦大学语料库的训练集和测试集(www.nlp.org.cn/docs/doclist.php? cat_id=16 & type=15)。选择 C32-Agriculture 和 C38-Politics 两类文本作为本文的训练集和测试集。在本实验中,进行 5 次增量学习,每次增量学习增加 400 篇训练集。

本实验采用式(2)构建 VSM 及其对应的文本特征向量,并且利用信息增益的特征提取方法进行特征提取。最后利用台湾大学林智仁副教授开发的 LIBSVM 的 Java 版本软件包进行文本分类。

5.2　增量学习实验结果及其分析

为了比较 FVI-SVM,分别与 I-SVM 和 VI-SVM 比较训练时间、训练产生 SV 数目、分类时间以及分类精度。

5.2.1　训练效率比较

图 2 比较了 I-SVM、VI-SVM 和 FVI-SVM 之间的训练效率。从图中可以看出:

a)随着增量学习的进行,I-SVM 的训练时间一直很高,这是因为 I-SVM 需要将整个增量训练集和历史训练集的并集作为 SVM 当前训练集。由于训练 SVM 分类器时间复杂度是

$O(n^3)$,因此该算法训练效率最低。

b)FVI-SVM 和 VI-SVM 训练时间较低,这主要是因为仅将增量训练集中违反 KKT 条件的文本特征向量加入当前训练集。虽然可能损失部分支持向量,但绝大多数非支持向量没有被加入当前训练集,因此算法分类效率比 I-SVM 高。

c)VI-SVM 的训练效率低于 FVI-SVM,这主要是因为冗余支持向量的存在,导致增量学习训练效率下降。

图 3 比较了 I-SVM、VI-SVM 和 FVI-SVM 之间的支持向量数目,从图中可以看出:

a)I-SVM 的支持向量数目高于 VI-SVM 和 FVI-SVM,这主要是因为后者在增量学习过程中,仅把违反 KKT 条件的文本特征向量加入当前训练集,损失掉了部分支持向量。

b)FVI-SVM 在后续增量学习过程中,支持向量数目保持在平稳状态,这主要是因为该算法修剪支持向量,控制了支持向量规模。

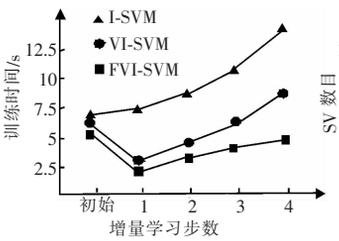

图 2 I-SVM、VI-SVM 和 FVI-SVM 训练效率比较

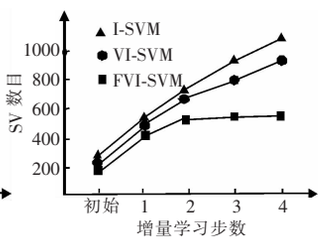

图 3 I-SVM、VI-SVM 和 FVI-SVM 中支持向量数目比较

5.2.2 分类性能比较

图 4 比较了 I-SVM、VI-SVM 和 FVI-SVM 之间的分类效率比较。从图中可以看出:

a)随着增量学习的进行,I-SVM 和 VI-SVM 的分类时间持续增加,这主要是因为在 SVM 分类过程中,每个测试文本特征向量都需要与全部支持向量进行点积运算。随着增量学习的进行,支持向量规模越来越大,因此分类效率越来越低。

b)FVI-SVM 可以保持平稳的分类效率,这主要是因为该算法修剪支持向量,控制了支持向量规模。

图 5 比较了 I-SVM、VI-SVM 和 FVI-SVM 之间的分类精度。从图中可以看出:

a)I-SVM 的分类精度高于 VI-SVM 和 FVI-SVM,这主要是因为 I-SVM 在增量学习过程中基本没有支持向量损失,而 VI-SVM 和 FVI-SVM 仅将违反 KKT 条件的特征向量加入当前训练集,有支持向量损失。

b)随着增量学习的进行, I-SVM、VI-SVM 和 FVI-SVM 的分类精度不断提高,这主要是因为在增量学习过程中,随着增量训练集的不断加入,分类信息不断得到完善。

c)FVI-SVM 的分类精度低于 VI-SVM,这主要是因为采用修剪冗余支持向量会导致最优分类超平面发生微量偏移。

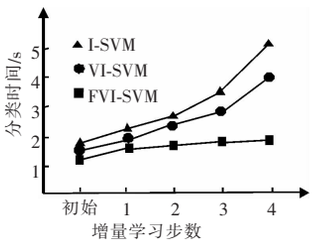

图 4 I-SVM、VI-SVM 和 FVI-SVM 中分类效率比较

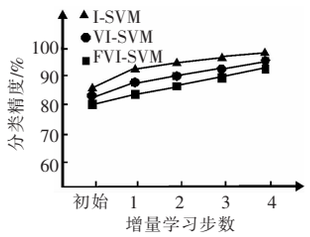

图 5 I-SVM、VI-SVM 和 FVI-SVM 之间的分类精度比较

6 结束语

FVI-SVM 将不违反 KKT 条件的增量文本特征向量剔除并通过 SNN 相似度剪除冗余支持向量,降低了历史训练集和当前训练集的规模,提高了增量学习的训练和分类效率,解决了由于 I-SVM 中缺乏增量训练集淘汰机制和冗余支持向量存在而导致训练效率以及分类效率下降的问题。实验表明 FVI-SVM 的训练效率和分类效率高于 I-SVM,并且随着增量学习的进行保持相对稳定。然而随着增量学习的进行,可能会产生不平衡分类问题,这是需要进一步解决的问题。

参考文献:

[1] PRONOBIS A, LUO Jie, CAPUTO B. The more you learn, the less you store: memory-controlled incremental SVM for visual place recognition [J]. Image and Vision Computing, 2010, 28(7): 1080-1097.

[2] DUAN Hua, SHAO Xiao-jian, HOU Wei-zhen, *et al*. An incremental learning algorithm for Lagrangian support vector machines [J]. Pattern Recognition Letters, 2009, 30(15): 1384-1391.

[3] 吴崇明,王晓丹,白冬婴,等. 基于类边界壳向量的快速 SVM 增量学习算法[J]. 计算机工程与应用,2010,46(23):185-187,248.

[4] YI Yang, WU Jian-sheng, XU Wei. Incremental SVM based on reserved set for network intrusion detection[J]. Experts Systems with Applications, 2011, 38(6): 7698-7707.

[5] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing[J]. Communication of the ACM, 1975, 18(11): 613-620.

[6] NGUYEN D, HO T B. A bottom-up method for simplifying support vector solutions [J]. IEEE Trans on Neural Networks, 2006, 17(3): 792-796.

(上接第 1244 页)

[6] 郭梅,朱金福. 基于模糊粗糙集的物流服务供应链绩效评价[J]. 系统工程,2007(7):48-52.

[7] KUMARA M, VRATB P, SHANKARC R. A fuzzy goal programming approach for vendor selection problem in a supply chain[J]. Computers & Industrial Engineering, 2004, 46(3): 69-85.

[8] HANNE N. The balanced scorecard: what is the score? A rhetorical analysis of the balanced scorecard[J]. Organizations and Society, 2003, 28(6): 591-619.

[9] 赵丽娟,罗兵. 绿色供应链中环境管理绩效模糊综合评价[J]. 重庆大学学报,2003,26(11):155-158.

[10] 吴隽,王兰义,李一军. 基于模糊质量功能展开的物流服务供应商选择研究[J]. 中国软科学,2010(3):145-151.

[11] PRTM. SCOR model [EB/OL]. (2002-09-10). http://www.ie.utoronto.ca/EIL/profiles/rune.

[12] GUNASEKARAN A, PATEL C, McCRAUGHEY R E. A framework for supply chain performance measurement[J]. International Journal of Production Economics, 2004, 87(3): 333-347.

[13] 霍佳震. 企业评价创新——集成化供应链绩效及其评价[M]. 石家庄:河北人民出版社,2001.

[14] BOLCH G, GREINER S, MEE H D, *et al*. Queuing networks and Markov chains: modeling and performance evaluation with computer science applications [M]. 2nd ed. New York: Wiley-Interscience, 2006.